

Obiekt Number

Dodał Administrator
poniedziałek, 15 marzec 2010 23:31

Dostępność: od JavaScript 1.1 i JScript 1.0

Przeglądarki: IE3, FF, NN3, OP7

Obiekt Number pozwala na przechowywanie wartości liczbowych oraz na prezentowanie ich z określoną precyzją. Zawiera również kilka właściwości reprezentujących wartości specjalne (np. określające nieskończoność). Wszystkie właściwości (opisane w kolejnym punkcie) są statyczne, czyli można się do nich odwoływać bez tworzenia obiektów typu Number. Samo utworzenie obiektu jest natomiast osiągnięte przez wywołanie w postaci:

```
new Number(wartość);
```

np.:

```
var liczba = new Number(100);
```

Właściwości obiektu Number

Właściwości udostępniane przez obiekty typu Number zostały przedstawione w tabeli 3.6. Są one dostępne we wszystkich popularnych przeglądarkach (IE4, FF, NN3, OP7).

Tabela 3.6. Właściwości udostępniane przez obiekty typu Number

Nazwa

Znaczenie

Wartość

Dostępność w JavaScriptcie

Dostępność w JScriptcie

MAX_VALUE

Określa maksymalną wartość liczbową, jaka może być reprezentowana w danej wersji JavaScriptu.

zazwyczaj 1.79769E+308

od wersji 1.1

od wersji 2.0

MIN_VALUE

Określa minimalną dodatnią wartość liczbową, jaka może być reprezentowana w danej wersji JavaScriptu.

zazwyczaj 5.0E-324

od wersji 1.1

od wersji 2.0

Obiekt Number

Dodał Administrator
poniedziałek, 15 marzec 2010 23:31

NaN

Wartość specjalna określająca, że dane wyrażenie nie reprezentuje wartości liczbowej.

literał NaN

od wersji 1.1

od wersji 2.0

NEGATIVE_INFINITY

Wartość specjalna określająca ujemną nieskończoność.

literał -Infinity

od wersji 1.1

od wersji 2.0

POSITIVE_INFINITY

Wartość specjalna określająca dodatnią nieskończoność.

literał Infinity

od wersji 1.1

od wersji 2.0

Metody obiektu Number

Metoda toExponential

Składnia: `number.toExponential([digits])`

Dostępność: od JavaScript 1.5 i JScript 5.5

Przeglądarki: IE5.5, FF, NN6, OP7

Metoda `toExponential` zwraca wartość reprezentowaną przez obiekt `number` w notacji wykładniczej. Opcjonalny argument `digits` określa liczbę miejsc po przecinku. Przykładowo: jeśli zostanie utworzony obiekt `Number` w postaci:

```
var number = new Number(49.987654321);
```

to wywołanie `number.toExponential(4)` da w wyniku `4.9988e+1`, a wywołanie `number.toExponential(6)` da w wyniku `4.998765e+1`. **Metoda toFixed**

Składnia: `number.toFixed([digits])`

Obiekt Number

Dodał Administrator
poniedziałek, 15 marzec 2010 23:31

Dostępność: od JavaScript 1.5 i JScript 5.5

Przeglądarki: IE, NN, OP, FF

Metoda `toFixed` zwraca wartość reprezentowaną przez obiekt `number` w notacji dziesiętnej. Opcjonalny argument `digits` określa liczbę miejsc po przecinku. Przykładowo: jeśli zostanie utworzony obiekt `Number` w postaci:

```
var number = new Number(49.987654321);
```

to wywołanie `number.toFixed(2)` da w wyniku 4.99, a wywołanie `number.toFixed(5)` da w wyniku 49.98765.

Metoda `toLocaleString`

Składnia: `number.toLocaleString()`

Dostępność: od JavaScript 1.5 i JScript 5.5

Przeglądarki: IE5.5, FF, NN6, OP7

Metoda `toLocaleString` zwraca ciąg znaków reprezentujący daną wartość z uwzględnieniem ustawień lokalnych przeglądarki. Przykładowo: jeżeli utworzymy obiekt typu `Number` za pomocą instrukcji:

```
var number = new Number(49.987654321);
```

to wywołanie:

```
number.toLocaleString()
```

da w wyniku następujące ciągi:

-

49,99 — w przypadku przeglądarki Internet Explorer,

-

49,987654321 — w przypadku przeglądarki Firefox,

-

49.987654321 — w przypadku przeglądarki Opera.

Metoda `toPrecision`

Składnia: `number.toPrecision([precision])`

Dostępność: od JavaScript 1.5 i JScript 5.5

Przeglądarki: IE5.5, FF, NN6, OP7

Metoda `toPrecision` zwraca wartość reprezentowaną przez obiekt `number` z liczbą cyfr znaczących określoną przez parametr `precision`. Przykładowo: jeśli zostanie utworzony obiekt `Number` w postaci:

```
var number = new Number(49.987654321);
```

to wywołanie `number.toPrecision(5)` da w wyniku 49.988. **Metoda `toString`**

Obiekt Number

Dodał Administrator
poniedziałek, 15 marzec 2010 23:31

Składnia: `number.toString([podstawa])`

Dostępność: od JavaScript 1.1 i JScript 2.0

Przeglądarki: IE4, FF, NN4, OP7

Metoda `toString` zwraca wartość reprezentowaną przez obiekt `number` w postaci ciągu znaków. Opcjonalny argument `podstawa` pozwala na określenie podstawy systemu liczbowego, np. 2 — system dwójkowy, 8 — system ósemkowy, 16 — system szesnastkowy. Domyślnie stosowany jest system dziesiętny. Na przykład gdyby istniała potrzeba wyświetlenia 16 kolejnych liczb (od 0 do 15) w reprezentacji dziesiętnej, dwójkowej, ósemkowej i szesnastkowej, można zastosować w tym celu skrypt z listingu 3.21.

Listing 3.21. Wartość w różnych systemach liczbowych

```
<script type="text/javascript">
for(i = 0; i <= 15; i++){
    var number = new Number(i);
    document.write(number.toString() + " ");
    document.write(number.toString(2) + " ");
    document.write(number.toString(8) + " ");
    document.write(number.toString(16) + " ");
}
</script> Metoda valueOf
```

Składnia: `number.valueOf()`

Dostępność: od JavaScript 1.1 i JScript 2.0

Przeglądarki: IE4, FF, NN3, OP7

Metoda `valueOf` zwraca podstawową wartość reprezentowaną przez obiekt `number`, czyli przechowywaną wartość liczbową.